INEXACT NEWTON-TYPE OPTIMIZATION WITH ITERATED SENSITIVITIES*

RIEN QUIRYNEN†, SÉBASTIEN GROS‡, AND MORITZ DIEHL§

Abstract. This paper presents and analyzes an Inexact Newton-type optimization method based on Iterated Sensitivities (INIS). A particular class of Nonlinear Programming (NLP) problems is considered, where a subset of the variables is defined by nonlinear equality constraints. The proposed algorithm considers any problem-specific approximation for the Jacobian of these constraints. Unlike other inexact Newton methods, the INIS-type optimization algorithm is shown to preserve the local convergence properties and the asymptotic contraction rate of the Newton-type scheme for the feasibility problem, yielded by the same Jacobian approximation. The INIS approach results in a computational cost which can be made close to that of the standard inexact Newton implementation. In addition, an adjoint-free (AF-INIS) variant of the approach is presented which, under certain conditions, becomes considerably easier to implement than the adjoint based scheme. The applicability of these results is motivated, specifically for dynamic optimization problems. In addition, the numerical performance of a specific open-source implementation is illustrated.

Key words. Newton-type methods, Optimization algorithms, Direct optimal control, Collocation methods

AMS subject classifications. 49M15, 90C30, 65M70

1. Introduction.

The present paper considers Newton-type optimization algorithms [19] for a class of Nonlinear Programming (NLP) problems

$$\min_{z,w} \quad f(z,w) \tag{1a}$$

$$\text{s.t.} \quad g(z,w) = 0, \tag{1b}$$

$$h(z,w) = 0, \tag{1c}$$

where $z \in \mathbb{R}^{n_\mathrm{z}}$ and $w \in \mathbb{R}^{n_\mathrm{w}}$ are the optimization variables. The objective and constraint functions are defined as $f : \mathbb{R}^{n_\mathrm{z}} \times \mathbb{R}^{n_\mathrm{w}} \to \mathbb{R}$, $g : \mathbb{R}^{n_\mathrm{z}} \times \mathbb{R}^{n_\mathrm{w}} \to \mathbb{R}^{n_g}$ and $h : \mathbb{R}^{n_\mathrm{z}} \times \mathbb{R}^{n_\mathrm{w}} \to \mathbb{R}^{n_\mathrm{h}}$ respectively, and are assumed to be twice continuously differentiable in all arguments. The subset of the variables z and the constraint function $g(\cdot)$ is selected such that $n_\mathrm{g} = n_\mathrm{z}$ and such that the Jacobian $\frac{\partial g(z,w)}{\partial z} \in \mathbb{R}^{n_\mathrm{z} \times n_\mathrm{z}}$ is invertible. It follows that the variables z are implicitly defined as functions of w via the nonlinear equality constraints $g(z,w) = 0$. This set of constraints in Eq. (1b) will be referred to throughout the paper as the *forward problem*, which delivers $z^\star(\bar{w})$ by solving the corresponding system

$$g(z, \bar{w}) = 0, \qquad \text{for a given value } \bar{w}. \tag{2}$$

Some interesting examples of such problem formulations result from a simultaneous approach to dynamic optimization [2, 10], where the forward problem imposes the

*This research was supported by the EU via ERC-HIGHWIND (259 166), FP7-ITN-TEMPO (607 957), H2020-ITN-AWESCO (642 682), by the DFG in context of the Research Unit FOR 2401 and by the Freiburg Institute for Advanced Studies (FRIAS). R. Quirynen holds a PhD fellowship of the Research Foundation – Flanders (FWO).

†Department ESAT-STADIUS, KU Leuven University, Kasteelpark Arenberg 10, 3001 Leuven, Belgium (rien.quirynen@esat.kuleuven.be).

‡Department of Signals and Systems, Chalmers University of Technology, Hörsalsvägen 11, SE-412 96 Göteborg, Sweden.

§Department IMTEK, University of Freiburg, Georges-Koehler-Allee 102, 79110 Freiburg, Germany.

system dynamics and therefore typically corresponds to a numerical simulation of differential equations. A popular example of such an approach is direct collocation [5], where the forward problem consists of the collocation equations and possibly including also the continuity conditions.

We are interested in solving the forward problem in Eq. (2) using Newton-type schemes that do not rely on an exact factorization of $g_z := \frac{\partial g}{\partial z}$, but use instead a full-rank approximation $M \approx g_z$. This Jacobian approximation can be used directly in a Newton-type method to solve the forward problem by steps

$$\Delta z = -M^{-1} g(\bar{z}, \bar{w}), \tag{3}$$

where $\bar{z}$ denotes the current guess and the full-step update in each Newton-type iteration can be written as $\bar{z}^+ = \bar{z} + \Delta z$. Even though local convergence properties for Newton-type optimization have been studied extensively in [6, 16, 19, 36], this paper presents a novel contribution regarding the connection between the accuracy of the Jacobian approximation M and the local contraction rate of the corresponding optimization algorithm. A Newton-type method with inexact derivatives does not converge to a solution of the original nonlinear optimization problem, unless adjoint derivatives are evaluated in order to compute the correct gradient of the Lagrangian [9, 22]. It has been pointed out by [39] that the locally linear convergence rate of the resulting Inexact Newton (IN) based optimization scheme is not strongly connected to the contraction of the iterations for the inner forward problem in (3). More specifically, it is possible that the Jacobian approximation M results in a fast contraction of the forward problem alone, while the optimization algorithm based on the same Jacobian approximation diverges. In contrast, the proposed INIS algorithm will be shown to have the same asymptotic contraction rate.

In this work, for the sake of simplicity, we omit (possibly nonlinear) inequality constraints in the NLP (1). Note however that our discussion on the local convergence of Newton-type optimization methods can be readily extended to the general case of inequality-constrained optimization. Such an extension can be based on techniques from Sequential Quadratic Programming (SQP) where, under mild conditions, the active set can be shown to be locally stable for the subproblems [9, 11, 38]. Hence, for the purpose of a local convergence analysis, the equality constraints in Eq. (1c) could additionally comprise the locally active inequality constraints. This observation is illustrated further in the numerical case study of this paper. Alternatively, the extension to inequality-constrained optimization can similarly be carried out in the context of interior-point methods [5, 37]. Convergence results for Newton-type optimization based on inexact derivative information can be found in [30, 34] for SQP or in [4, 48] for nonlinear Interior Point (IP) methods. An alternative approach makes use of inexact solutions to the linearized subproblems in order to reduce the overall computational burden of the Newton-type scheme as discussed in [14, 15, 30, 35]. Note that other variants of inexact Newton-type algorithms exist, e.g., allowing locally superlinear convergence [22, 29] based on quasi-Newton Jacobian updates. In the case of optimal control for differential-algebraic equations, even quadratic convergence rates [31] have been observed under certain conditions.

1.1. Contributions and Outline. The main contribution of the present paper is the Inexact Newton method with Iterated Sensitivities (INIS) that allows one to recover a strong connection between the local contraction rate of the forward problem and the local convergence properties of the resulting Newton-type optimization algorithm. More specifically, local contraction based on the Jacobian approximation for the forward problem is necessary, and under mild conditions even sufficient for local convergence of the INIS-type optimization scheme. The article presents an efficient implementation of the INIS algorithm, resulting in a computational cost close to that of the standard inexact Newton implementation. Note that this Newton-type scheme shows a particular resemblance to the lifted Newton method in [1], based on a lifting of the forward sensitivities. This connection is also discussed in [44], in the context of collocation methods for direct optimal control.

In addition, an adjoint-free (AF-INIS) variant for Newton-type optimization is proposed. This alternative approach can be interesting whenever the algorithm can be carried out independently of the respective values for the multipliers corresponding to the equality constraints, but it generally does not preserve the local convergence properties of the forward scheme. As discussed also further, an adjoint-free implementation can however be attractive in case of a sequence of nontrivial operations, e.g., resulting from a numerical simulation in dynamic optimization. An open-source implementation of these novel INIS-type techniques for simultaneous direct optimal control is proposed as part of the `ACADO` Toolkit. Throughout the article, theoretical results are illustrated using toy examples of quadratic and nonlinear programming problems. In addition, the numerical performance of the open-source implementation is shown on the benchmark case study of the optimal control for a chain of masses.

The paper is organized as follows. Section 2 briefly presents standard Newton-type optimization methods. Section 3 then proposes and analyzes the Inexact Newton method based on Iterated Sensitivities (INIS) as an alternative implementation of inexact Newton-type optimization. An adjoint-free variant of the INIS-type optimization algorithm is presented in Section 4. An important application of the proposed schemes for simultaneous approaches of direct optimal control is presented in Section 5, including numerical results based on a specific open-source implementation. Section 6 finally concludes this paper.

2. Newton-Type Optimization.

The Lagrange function for the NLP (1) reads as $\mathcal{L}(y,\lambda) = f(y) + \mu^\top g(y) + \nu^\top h(y)$, where $y := \begin{bmatrix} z^\top & w^\top \end{bmatrix}^\top \in \mathbb{R}^{n_\mathrm{y}}$ denotes all primal variables. In addition, $c(y) := \begin{bmatrix} g(y)^\top & h(y)^\top \end{bmatrix}^\top$, $\lambda := \begin{bmatrix} \mu^\top & \nu^\top \end{bmatrix}^\top \in \mathbb{R}^{n_\mathrm{c}}$ is defined and $n_\mathrm{c} = n_\mathrm{g} + n_\mathrm{h}$, where $\mu \in \mathbb{R}^{n_\mathrm{g}}$, $\nu \in \mathbb{R}^{n_\mathrm{h}}$ respectively denote the multipliers for the nonlinear equality constraints in Eqs. (1b) and (1c). The first-order necessary conditions for optimality are then defined as

$$\begin{aligned} &\nabla_y \mathcal{L}(y,\lambda): \qquad && \nabla_y f(y) + \sum_{i=1}^{n_\mathrm{c}} \lambda_i \nabla_y c_i(y) = 0, \\ &\nabla_\lambda \mathcal{L}(y,\lambda): && c(y) = 0, \end{aligned} \tag{4}$$

and are generally referred to as the Karush-Kuhn-Tucker (KKT) conditions [37]. Note that a more compact notation is used to denote the gradient of a scalar function, i.e., this is the transpose of the Jacobian $\nabla_y \mathcal{L}(\cdot) = \frac{\partial \mathcal{L}}{\partial y}(\cdot)^\top = \mathcal{L}_y(\cdot)^\top$. We further generalize this operator as $\nabla_y c(\cdot) = \begin{bmatrix} \nabla_y c_1(\cdot) & \cdots & \nabla_y c_{n_\mathrm{c}}(\cdot) \end{bmatrix} = \frac{\partial c}{\partial y}(\cdot)^\top = c_y(\cdot)^\top$. This nonlinear system of equations can also be written in the compact notation

$$\mathcal{F}(y,\lambda) = \begin{bmatrix} \nabla_y \mathcal{L}(y,\lambda) \\ c(y) \end{bmatrix} = 0. \tag{5}$$

Each local minimizer $(y^\star,\lambda^\star)$ of the NLP (1) is assumed to be a regular KKT point $\mathcal{F}(y^\star,\lambda^\star) = 0$ as defined next. For this purpose, we rely on the linear independence constraint qualification (LICQ) and the second order sufficient conditions (SOSC) for optimality, of which the latter requires that the Hessian of the Lagrangian is strictly positive definite in the directions of the critical cone [37].

DEFINITION 1. *A minimizer of an equality constrained NLP is called a regular KKT point, if both LICQ and SOSC are satisfied at this KKT point.*

2.1. Newton-Type Methods. Newton-type optimization proceeds with applying a variant of Newton's method [17, 19] to find a solution to the KKT system in Eq. (5). Note that an exact Newton iteration on the KKT conditions reads as:

$$\underbrace{\begin{bmatrix} \nabla_y^2 \mathcal{L}(\bar{y},\bar{\lambda}) & c_y^\top(\bar{y}) \\ c_y(\bar{y}) & \mathbb{0} \end{bmatrix}}_{=J(\bar{y},\bar{\lambda})} \begin{bmatrix} \Delta y \\ \Delta \lambda \end{bmatrix} = - \underbrace{\begin{bmatrix} \nabla_y \mathcal{L}(\bar{y},\bar{\lambda}) \\ c(\bar{y}) \end{bmatrix}}_{=\mathcal{F}(\bar{y},\bar{\lambda})}, \tag{6}$$

where $c_y(\bar{y}) := \frac{\partial c}{\partial y}(\bar{y})$ denotes the Jacobian matrix and the values $\bar{y}$ and $\bar{\lambda}$ denote the primal and dual variables at the current guess. In the following, we will refer to the exact Newton iteration using the compact notation:

$$J(\bar{y},\bar{\lambda}) \begin{bmatrix} \Delta y \\ \Delta \lambda \end{bmatrix} = -\mathcal{F}(\bar{y},\bar{\lambda}) \tag{7}$$

where the exact Jacobian matrix is defined as $J(\bar{y},\bar{\lambda}) := \frac{\partial \mathcal{F}}{\partial (y,\lambda)}(\bar{y},\bar{\lambda})$. In this work, a full-step update of the primal and dual variables is considered for simplicity in each iteration, i.e., $\bar{y}^+ = \bar{y} + \Delta y$ and $\bar{\lambda}^+ = \bar{\lambda} + \Delta\lambda$ even though globalization strategies are typically used to guarantee convergence [5, 37].

As mentioned earlier, many Newton-type optimization methods have been proposed that result in desirable local convergence properties at a considerably reduced computational cost by either forming an approximation of the KKT matrix $J(\cdot)$ or by solving the linear system (6) approximately [35]. For example, the family of quasi-Newton methods [18, 37] is based on the approximation of the Hessian of the Lagrangian $\tilde{H} \approx H := \nabla_y^2 \mathcal{L}$ using only first order derivative information. Other Newton-type optimization algorithms even use an inexact Jacobian for the nonlinear constraints [9, 22, 31, 47] as discussed next.

2.2. Adjoint-Based Inexact Newton (IN). Let us consider the invertible Jacobian approximation $M \approx g_z$ in the Newton-type method of Eq. (3) to solve the forward problem. The resulting inexact Newton method aimed at solving the KKT conditions for the NLP in Eq. (5), iteratively solves the following linear system

$$\underbrace{\begin{bmatrix} \tilde{H} & \begin{pmatrix} M^\top & h_z^\top \\ g_w^\top & h_w^\top \end{pmatrix} \\ \begin{pmatrix} M & g_w \\ h_z & h_w \end{pmatrix} & \mathbb{0} \end{bmatrix}}_{=: \tilde{J}_{\mathrm{IN}}(\bar{y},\bar{\lambda})} \begin{bmatrix} \Delta z \\ \Delta w \\ \Delta \mu \\ \Delta \nu \end{bmatrix} = - \begin{bmatrix} \nabla_y \mathcal{L}(\bar{y},\bar{\lambda}) \\ c(\bar{y}) \end{bmatrix}, \tag{8}$$

where the right-hand side of the system is exact as in Eq. (6) and an approximation of the Hessian $\tilde{H} \approx \nabla_y^2 \mathcal{L}(\bar{y},\bar{\lambda})$ has been introduced for the sake of completeness. Note that the gradient of the Lagrangian $\nabla_y \mathcal{L}(\cdot)$ can be evaluated efficiently using adjoint differentiation techniques, such that the scheme is often referred to as an *adjoint-based Inexact Newton* (IN) method [9, 22]. The corresponding convergence analysis will be discussed later. Algorithm 1 describes an implementation to solve the adjoint-based IN system in (8). It relies on a numerical elimination of the variables $\Delta z = -M^{-1}(g(\bar{y}) + g_w \Delta w)$ and $\Delta\mu$ such that a smaller system is solved in the variables $\Delta w, \Delta\nu$, which can be expanded back into the full variable space. Note that one recovers the Newton-type iteration on the forward problem $\Delta z = -M^{-1}g(\bar{y})$ for a fixed value $\bar{w}$, i.e., in case $\Delta w = 0$.

Algorithm 1 One iteration of an adjoint-based Inexact Newton (IN) method.

Input: Current values $\bar{y} = (\bar{z}, \bar{w})$, $\bar{\lambda} = (\bar{\mu}, \bar{\nu})$ and approximations M, $\tilde{H}(\bar{y}, \bar{\lambda})$.

1: After eliminating the variables Δz, $\Delta\mu$ in (8), solve the resulting system:
$$\begin{bmatrix} \tilde{Z}^\top \tilde{H} \tilde{Z} & \tilde{Z}^\top h_y^\top \\ h_y \tilde{Z} & \mathbb{0} \end{bmatrix} \begin{bmatrix} \Delta w \\ \Delta\nu \end{bmatrix} = - \begin{bmatrix} \tilde{Z}^\top \nabla_y \mathcal{L}(\bar{y}, \bar{\lambda}) \\ h(\bar{y}) \end{bmatrix} - \begin{bmatrix} \tilde{Z}^\top \tilde{H} \\ h_y \end{bmatrix} \begin{bmatrix} -M^{-1} g(\bar{y}) \\ \mathbb{0} \end{bmatrix},$$
where $\tilde{Z}^\top := \left[-g_w^\top M^{-\top},\ \mathbb{1}_{n_\mathrm{w}}\right]$.

2: Based on Δw and $\Delta\nu$, the corresponding values for Δz and $\Delta\mu$ are found:
$\Delta z = -M^{-1}(g(\bar{y}) + g_w \Delta w)$ and
$\Delta\mu = -\begin{bmatrix} M^{-\top} & \mathbb{0} \end{bmatrix} \left(\nabla_y \mathcal{L}(\bar{y}, \bar{\lambda}) + \tilde{H} \Delta y + h_y^\top \Delta\nu \right)$.

Output: New values $\bar{y}^+ = \bar{y} + \Delta y$ and $\bar{\lambda}^+ = \bar{\lambda} + \Delta\lambda$.

REMARK 2. *The matrix $\tilde{Z}^\top := \left[-g_w^\top M^{-\top},\ \mathbb{1}_{n_\mathrm{w}}\right]$ in step 1 of Algorithm 1 is an approximation for $Z^\top := \left[-g_w^\top g_z^{-\top},\ \mathbb{1}_{n_\mathrm{w}}\right]$, which denotes a basis for the null space of the constraint Jacobian $g_y Z = \mathbb{0}$ such that $Z^\top \nabla_y \mathcal{L}(\bar{y}, \bar{\lambda}) = Z^\top (\nabla_y f(\bar{y}) + \nabla_y h(\bar{y})\bar{\nu})$. When using instead the approximate matrix $\tilde{Z}$, this results in the following correction of the gradient term*

$$\tilde{Z}^\top \nabla_y \mathcal{L}(\bar{y}, \bar{\lambda}) = \tilde{Z}^\top (\nabla_y f(\bar{y}) + \nabla_y h(\bar{y})\bar{\nu}) - \left((g_z M^{-1} - \mathbb{1}_{n_\mathrm{z}}) g_w \right)^\top \bar{\mu}. \tag{9}$$

2.3. Newton-Type Local Convergence. One iteration of the adjoint-based IN method solves the linear system in Eq. (8), which can be written in the following compact form

$$\tilde{J}_{\mathrm{IN}}(\bar{y}, \bar{\lambda}) \begin{bmatrix} \Delta y \\ \Delta\lambda \end{bmatrix} = -\mathcal{F}(\bar{y}, \bar{\lambda}), \tag{10}$$

where $\mathcal{F}(\cdot)$ denotes the exact KKT right-hand side in Eq. (6). The convergence of this scheme then follows the classical and well-known local contraction theorem from [6, 19, 22, 39]. We use a particular version of this theorem from [20], providing sufficient and necessary conditions for the existence of a neighborhood of the solution where the Newton-type iteration converges. Let $\rho(P)$ denote the spectral radius, i.e., the maximum absolute value of the eigenvalues for the square matrix P.

THEOREM 3 (Local Newton-type contraction [20]). *We consider the twice continuously differentiable function $\mathcal{F}(y, \lambda)$ from Eq. (5) and the regular KKT point $\mathcal{F}(y^\star, \lambda^\star) = 0$ from Definition 1. We then apply the Newton-type iteration in Eq. (10), where $\tilde{J}_{\mathrm{IN}}(\bar{y}, \bar{\lambda}) \approx J(\bar{y}, \bar{\lambda})$ is additionally assumed to be continuously differentiable and invertible in a neighborhood of the solution. If all eigenvalues of the iteration matrix have a modulus smaller than one, i.e., if the spectral radius satisfies*

$$\kappa^\star := \rho \left(\tilde{J}_{\mathrm{IN}}(y^\star, \lambda^\star)^{-1} J(y^\star, \lambda^\star) - \mathbb{1}_{n_\mathcal{F}} \right) < 1, \tag{11}$$

then this fixed point $(y^\star, \lambda^\star)$ is asymptotically stable, where $n_{\mathcal{F}} = n_{\mathrm{y}} + n_{\mathrm{g}} + n_{\mathrm{h}}$. Additionally, the iterates $(\bar{y}, \bar{\lambda})$ converge linearly to the KKT point $(y^\star, \lambda^\star)$ with the asymptotic contraction rate $\kappa^\star$ when initialized sufficiently close. On the other hand, if $\kappa^\star > 1$, then the fixed point $(y^\star, \lambda^\star)$ is unstable.

A proof for Theorem 3 can be found in [20, 41], based on a classical stability result from nonlinear systems theory.

REMARK 4. *The inexact Newton method relies on the standard assumption that the Jacobian and Hessian approximations M and $\tilde{H}$ in (8) are such that the corresponding matrix $\tilde{J}_{\mathrm{IN}}(\bar{y}, \bar{\lambda})$ is invertible. However, in addition, Theorem 3 requires that $\tilde{J}_{\mathrm{IN}}(\cdot)$ is continuously differentiable in a neighborhood of the solution. This assumption is satisfied, e.g., for an exact Newton method, for fixed Jacobian approximations as well as for the Generalized Gauss-Newton (GGN) method for least squares type optimization [8, 20]. This theorem on local Newton-type convergence will therefore be sufficient for our discussion, even though more advanced results exist [16, 19].*

REMARK 5. *As mentioned earlier in the introduction, an inequality constrained problem can be solved with any of the proposed Newton-type optimization algorithms, in combination with techniques from either SQP or interior point methods to treat the inequality constraints. Let us consider a local minimizer which is assumed to be regular, i.e., it satisfies the linear independence constraint qualification (LICQ), the strict complementarity condition and the second order sufficient conditions (SOSC) as defined in [37]. In this case, the primal-dual central path associated with this minimizer is locally unique when using an interior point method. In case of an SQP method, under mild conditions on the Hessian and Jacobian approximations, the corresponding active set is locally stable in a neighborhood of the minimizer, i.e., the solution of each QP subproblem has the same active set as the original NLP [9, 11, 46]. Hence, for the purpose of a local convergence analysis, the equality constraints in Eq.* (1c) *could additionally comprise the active inequality constraints in a neighborhood of the local minimizer.*

2.4. A Motivating QP Example. In this paper, we are interested in the existence of a connection between the Newton-type iteration on the forward problem (3) being locally contractive, i.e. $\kappa^\star_{\mathrm{F}} := \rho\left(M^{-1} g_z(z^\star, \bar{w}) - \mathbb{1}_{n_{\mathrm{z}}}\right) < 1$, and the local convergence for the corresponding Newton-type optimization algorithm as defined by Theorem 3. From the detailed discussion in [7, 39, 40], we know that contraction for the forward problem is neither sufficient nor necessary for convergence of the adjoint-based Inexact Newton (IN) type method in Algorithm 1, even when using an exact Hessian $H = \nabla_y^2 \mathcal{L}(\bar{y}, \bar{\lambda})$. To support this statement let us consider the following Quadratic Programming (QP) example from Potschka [39], based on a linear constraint $g(y) = \begin{bmatrix} A_1 & A_2 \end{bmatrix} y$, $n_{\mathrm{h}} = 0$ and quadratic objective $f(y) = \frac{1}{2} y^\top H y$ in (1). The matrix A_1 is assumed invertible and close to identity, such that we can select the Jacobian approximation $M = \mathbb{1}_{n_{\mathrm{z}}} \approx A_1$. The problem data from [39] read as

$$H = \begin{bmatrix} 0.83 & 0.083 & 0.34 & -0.21 \\ 0.083 & 0.4 & -0.34 & -0.4 \\ 0.34 & -0.34 & 0.65 & 0.48 \\ -0.21 & -0.4 & 0.48 & 0.75 \end{bmatrix}, \qquad (12)$$

$$A_1 = \begin{bmatrix} 1.1 & 1.7 \\ 0 & 0.52 \end{bmatrix}, \; A_2 = \begin{bmatrix} -0.55 & -1.4 \\ -0.99 & -1.8 \end{bmatrix}.$$

For this specific QP instance, we may compute the linear contraction rate for the

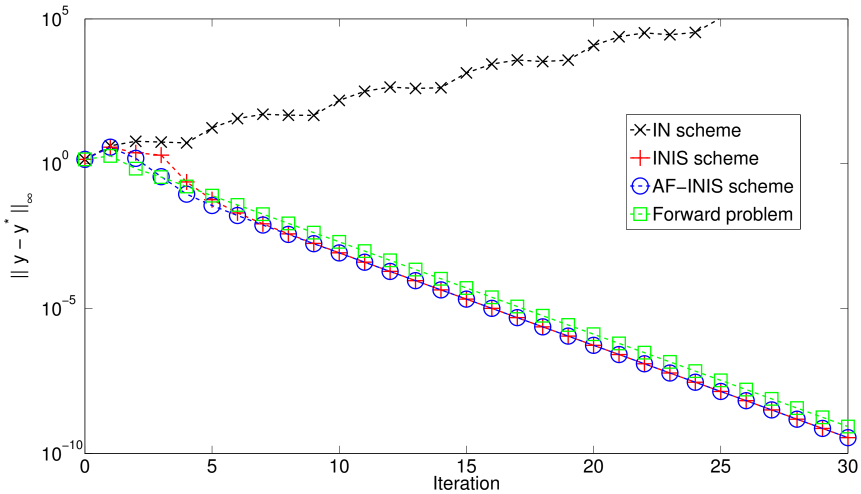

FIG. 1. *Illustration of the divergence of the Inexact Newton (IN) and the convergence of the Inexact Newton with Iterated Sensitivities (INIS) scheme for the QP in Eq. (12). In addition, the rate of convergence for INIS can be observed to be the same as for the forward problem.*

Newton-type method on the forward problem (3):

$$\kappa_{\mathrm{F}}^{\star} = \rho(M^{-1}g_z - \mathbb{1}_{n_z}) = \rho(A_1 - \mathbb{1}_{n_z}) = 0.48 < 1.$$

In addition, let us consider the IN algorithm based on the solution of the linear system (8) for the same QP example using the exact Hessian $\tilde{H} = H$. We can then compute the corresponding contraction rate at the solution point:

$$\kappa_{\mathrm{IN}}^{\star} = \rho(\tilde{J}_{\mathrm{IN}}^{-1} J - \mathbb{1}_{n_{\mathcal{F}}}) \approx 1.625 > 1,$$

where $J = J(y, \lambda)$ denotes the exact Jacobian of the KKT system in Eq. (5). For this QP example (12), the Newton-type method on the forward problem locally converges with $\kappa_{\mathrm{F}}^{\star} = 0.48 < 1$, while the corresponding IN algorithm is unstable $\kappa_{\mathrm{IN}}^{\star} \approx 1.625 > 1$. In what follows, we present and study a novel Newton-type optimization algorithm based on iterated sensitivities, which circumvents this problem at a negligible additional computational cost. These observations are illustrated in Figure 1, which presents the Newton-type iterations for the different algorithms, starting from the same initial point and using the same Jacobian approximation. The figure includes the linear convergence for the Newton-type method (3) on the forward problem.

3. Inexact Newton with Iterated Sensitivities (INIS). Let us introduce an alternative inexact Newton-type optimization algorithm, labelled INIS in the following, based on the solution of an augmented KKT system defined as

$$\mathcal{F}_{\mathrm{INIS}}(y, \lambda, D) = \begin{bmatrix} \nabla_y \mathcal{L}(y, \lambda) \\ c(y) \\ \mathrm{vec}(g_z D - g_w) \end{bmatrix} = 0, \tag{13}$$

where the additional variable $D \in \mathbb{R}^{n_z \times n_w}$ denotes the sensitivity matrix, implicitly defined by the equation $g_z D - g_w = 0$[1]. The number of variables in this augmented system is denoted by $n_{\mathrm{INIS}} = n_{\mathcal{F}} + n_{\mathrm{D}}$, where $n_{\mathcal{F}} = n_{\mathrm{y}} + n_{\mathrm{g}} + n_{\mathrm{h}}$ and

$n_{\mathrm{D}} = n_{\mathrm{z}} n_{\mathrm{w}}$. The following proposition states the connection between the augmented system $\mathcal{F}_{\mathrm{INIS}}(y, \lambda, D) = 0$ and the original KKT system in Eq. (5).

PROPOSITION 6. *A regular point $(y^\star, \lambda^\star, D^\star)$ for the augmented system in (13), corresponds to a regular KKT point $(y^\star, \lambda^\star)$ for the NLP in Eq. (1).*

Proof. This result follows directly from observing that the first two equations of the augmented system $\mathcal{F}_{\mathrm{INIS}}(y, \lambda, D) = 0$ correspond to the KKT conditions $\mathcal{F}(y, \lambda) = 0$ in (5) for the original NLP problem in Eq. (1). ∎

3.1. Implementation. We introduce the Inexact Newton method with Iterated Sensitivities (INIS), to iteratively solve the augmented KKT system in (13) based on

(14)

$$\underbrace{\begin{bmatrix} \tilde{H} & \begin{pmatrix} M^\top & h_z^\top \\ \bar{D}^\top M^\top & h_w^\top \end{pmatrix} & \mathbb{0} \\ \begin{pmatrix} M & M\,\bar{D} \\ h_z & h_w \end{pmatrix} & \mathbb{0} & \mathbb{0} \\ \mathbb{0} & \mathbb{0} & \mathbb{1}_{n_{\mathrm{w}}} \otimes M \end{bmatrix}}_{=:\, \tilde{J}_{\mathrm{INIS}}(\bar{y}, \bar{\lambda}, \bar{D})} \begin{bmatrix} \Delta z \\ \Delta w \\ \Delta \mu \\ \Delta \nu \\ \mathrm{vec}(\Delta D) \end{bmatrix} = - \underbrace{\begin{bmatrix} \nabla_y \mathcal{L}(\bar{y}, \bar{\lambda}) \\ c(\bar{y}) \\ \mathrm{vec}(g_z \bar{D} - g_w) \end{bmatrix}}_{= \mathcal{F}_{\mathrm{INIS}}(\bar{y}, \bar{\lambda}, \bar{D})},$$

where $\otimes$ denotes the Kronecker product of matrices, and where we use the Jacobian approximation $M \approx g_z$ from the Newton-type method on the forward problem in (3). The resulting matrix $\tilde{J}_{\mathrm{INIS}}(\bar{y}, \bar{\lambda}, \bar{D})$ forms an approximation for the exact Jacobian $J_{\mathrm{INIS}}(\bar{y}, \bar{\lambda}, \bar{D}) := \frac{\partial \mathcal{F}_{\mathrm{INIS}}}{\partial (y, \lambda, D)}(\bar{y}, \bar{\lambda}, \bar{D})$ of the augmented system. Similar to Remark 4, we assume that the Jacobian and Hessian approximations M, $\bar{D}$ and $\tilde{H}$ are such that the INIS matrix $\tilde{J}_{\mathrm{INIS}}(\cdot)$ is continuously differentiable and invertible.

Algorithm 2 shows that the INIS scheme in Eq. (14) can be implemented efficiently using a condensing and expansion procedure and the computational cost can be made close to that of the standard inexact Newton method in Algorithm 1. More specifically, the INIS scheme requires the linear system solution $-M^{-1}(g_z \bar{D} - g_w)$, for which the right-hand side can be evaluated efficiently using AD techniques [28]. Similar to Remark 2, let us write the gradient correction in step 1 of Algorithm 2:

$$\tilde{Z}^\top \nabla_y \mathcal{L}(\bar{y}, \bar{\lambda}) = \tilde{Z}^\top (\nabla_y f(\bar{y}) + \nabla_y h(\bar{y}) \bar{\nu}) - (g_z \bar{D} - g_w)^\top \bar{\mu}, \tag{15}$$

where $\tilde{Z}^\top := \left[-\bar{D}^\top,\ \mathbb{1}_{n_{\mathrm{w}}}\right]$. Note that the evaluation of $g_z \bar{D} - g_w$ can be reused in step 1 and 3 of Algorithm 2, which allows INIS to be computationally competitive with the standard IN scheme. This will also be illustrated by the numerical results for direct optimal control in Section 5.

3.2. Local Contraction Theorem. In what follows, we show that Algorithm 2 allows one to recover the connection between the contraction properties of the forward problem and the one of the Newton-type optimization algorithm. This observation makes the INIS-type optimization scheme depart fundamentally from the classical adjoint-based IN method. The local contraction of the forward problem will be shown to be necessary for the local convergence of the INIS algorithm, and can be sufficient under reasonable assumptions on the Hessian approximation $\tilde{H}$.

Let us formalize the local contraction rate $\kappa^\star_{\mathrm{INIS}} = \rho(\tilde{J}_{\mathrm{INIS}}^{-1} J_{\mathrm{INIS}} - \mathbb{1}_{n_{\mathrm{INIS}}})$ for the

[1]The operator vec(·) denotes a vectorization of a matrix, i.e., this is a linear transformation that converts the matrix into a column vector.

Algorithm 2 One iteration of an adjoint-based Inexact Newton with Iterated Sensitivities (INIS) optimization method.

Input: Current values $\bar{y} = (\bar{z}, \bar{w})$, $\bar{\lambda} = (\bar{\mu}, \bar{\nu})$, $\bar{D}$ and approximations M, $\tilde{H}(\bar{y}, \bar{\lambda})$.

1: After eliminating the variables Δz, $\Delta\mu$ in (14), solve the resulting system:
$$\begin{bmatrix} \tilde{Z}^\top \tilde{H} \tilde{Z} & \tilde{Z}^\top h_y^\top \\ h_y \tilde{Z} & \mathbb{0} \end{bmatrix} \begin{bmatrix} \Delta w \\ \Delta \nu \end{bmatrix} = - \begin{bmatrix} \tilde{Z}^\top \nabla_y \mathcal{L}(\bar{y}, \bar{\lambda}) \\ h(\bar{y}) \end{bmatrix} - \begin{bmatrix} \tilde{Z}^\top \tilde{H} \\ h_y \end{bmatrix} \begin{bmatrix} -M^{-1} g(\bar{y}) \\ \mathbb{0} \end{bmatrix},$$
where $\tilde{Z}^\top := \begin{bmatrix} -\bar{D}^\top, & \mathbb{1}_{n_\mathrm{w}} \end{bmatrix}$.

2: Based on Δw and $\Delta\nu$, the corresponding values for Δz and $\Delta\mu$ are found:
$\Delta z = -M^{-1} g(\bar{y}) - \bar{D} \Delta w$ and
$\Delta\mu = - \begin{bmatrix} M^{-\top} & \mathbb{0} \end{bmatrix} \left(\nabla_y \mathcal{L}(\bar{y}, \bar{\lambda}) + \tilde{H} \Delta y + h_y^\top \Delta \nu \right)$.

3: Independently, the sensitivity matrix is updated in each iteration:
$\Delta D = -M^{-1}(g_z \bar{D} - g_w)$.

Output: New values $\bar{y}^+ = \bar{y} + \Delta y$, $\bar{\lambda}^+ = \bar{\lambda} + \Delta\lambda$ and $\bar{D}^+ = \bar{D} + \Delta D$.

INIS scheme (14), where the Jacobian of the augmented KKT system (13) reads:

$$(16) \qquad J_{\mathrm{INIS}} = \begin{bmatrix} \nabla_y^2 \mathcal{L} & c_y^\top & \mathbb{0} \\ c_y & \mathbb{0} & \mathbb{0} \\ s_y & \mathbb{0} & \mathbb{1}_{n_\mathrm{w}} \otimes g_z \end{bmatrix} \quad \text{where } s_y := \frac{\partial}{\partial y} \mathrm{vec}(g_z D - g_w).$$

The following theorem specifies the eigenspectrum of the iteration matrix $\tilde{J}_{\mathrm{INIS}}^{-1} J_{\mathrm{INIS}} - \mathbb{1}_{n_{\mathrm{INIS}}}$ at the solution point $(y^\star, \lambda^\star, D^\star)$, using the notation $\sigma(P)$ to denote the spectrum, i.e., the set of eigenvalues for a matrix P.

THEOREM 7. *For the augmented linear system (14) on the NLP in Eq. (1), the eigenspectrum of the INIS-type iteration matrix at the solution $(y^\star, \lambda^\star, D^\star)$ reads as*

$$(17) \qquad \sigma\left(\tilde{J}_{\mathrm{INIS}}^{-1} J_{\mathrm{INIS}} - \mathbb{1}_{n_{\mathrm{INIS}}}\right) = \{0\} \cup \sigma\left(M^{-1} g_z - \mathbb{1}_{n_\mathrm{z}}\right) \cup \sigma\left(\tilde{H}_Z^{-1} H_Z - \mathbb{1}_{\tilde{n}_\mathrm{Z}}\right),$$

where $\tilde{n}_\mathrm{Z} = n_\mathrm{w} - n_\mathrm{h}$ and $Z \in \mathbb{R}^{n_\mathrm{y} \times \tilde{n}_\mathrm{Z}}$ denotes a basis for the null space of the complete constraint Jacobian c_y, such that the reduced Hessians $H_Z := Z^\top H Z \in \mathbb{R}^{\tilde{n}_\mathrm{Z} \times \tilde{n}_\mathrm{Z}}$ and $\tilde{H}_Z := Z^\top \tilde{H} Z \in \mathbb{R}^{\tilde{n}_\mathrm{Z} \times \tilde{n}_\mathrm{Z}}$ are defined. Note that $H := \nabla_y^2 \mathcal{L}(y^\star, \lambda^\star)$ is the exact Hessian and $\tilde{H} \approx H$ is an approximation. More specifically, the iteration matrix has the $\tilde{n}_\mathrm{Z}$ eigenvalues of the matrix $\tilde{H}_Z^{-1} H_Z - \mathbb{1}_{\tilde{n}_\mathrm{Z}}$, the n_z eigenvalues of $M^{-1} g_z - \mathbb{1}_{n_\mathrm{z}}$ with an algebraic multiplicity of $(2 + n_\mathrm{w})$ and $\{0\}$ with algebraic multiplicity $(2\, n_\mathrm{h})$.

Proof. At the solution of the augmented KKT system for the NLP in Eq. (13), the sensitivity matrix corresponds to $D^\star = g_z^{-1} g_w$. We then introduce the following Jacobian matrix and its approximation:

$$g_y = \begin{bmatrix} g_z & g_w \end{bmatrix}, \qquad \tilde{g}_y = \begin{bmatrix} \mathbb{1}_{n_\mathrm{z}} & D^\star \end{bmatrix} = g_z^{-1} g_y,$$

such that the exact and inexact augmented Jacobian matrices read

$$(18) \qquad J_{\mathrm{INIS}} = \begin{bmatrix} H & g_y^\top & h_y^\top & \mathbb{0} \\ g_y & \mathbb{0} & \mathbb{0} & \mathbb{0} \\ h_y & \mathbb{0} & \mathbb{0} & \mathbb{0} \\ s_y & \mathbb{0} & \mathbb{0} & \mathbb{1}_{n_\mathrm{w}} \otimes g_z \end{bmatrix}, \quad \tilde{J}_{\mathrm{INIS}} = \begin{bmatrix} \tilde{H} & \tilde{g}_y^\top M^\top & h_y^\top & \mathbb{0} \\ M\, \tilde{g}_y & \mathbb{0} & \mathbb{0} & \mathbb{0} \\ h_y & \mathbb{0} & \mathbb{0} & \mathbb{0} \\ \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{1}_{n_\mathrm{w}} \otimes M \end{bmatrix},$$

at the solution point $(y^\star, \lambda^\star, D^\star)$. We observe that the eigenvalues γ of the iteration matrix $\tilde{J}_{\text{INIS}}^{-1} J_{\text{INIS}} - \mathbb{1}_{n_{\text{INIS}}}$ are the zeros of

$$\det\left(\tilde{J}_{\text{INIS}}^{-1} J_{\text{INIS}} - \mathbb{1}_{n_{\text{INIS}}} - \gamma \mathbb{1}_{n_{\text{INIS}}}\right) = \det\left(\tilde{J}_{\text{INIS}}^{-1} J_{\text{INIS}} - (\gamma+1) \mathbb{1}_{n_{\text{INIS}}}\right) = 0.$$

Since $\tilde{J}_{\text{INIS}}$ is invertible, the second equality holds if and only if

$$\det\left(\tilde{J}_{\text{INIS}}\left(\tilde{J}_{\text{INIS}}^{-1} J_{\text{INIS}} - (\gamma+1) \mathbb{1}_{n_{\text{INIS}}}\right)\right) = \det\left(J_{\text{INIS}} - (\gamma+1)\tilde{J}_{\text{INIS}}\right) = 0.$$

Using the notation in Eq. (18), we can rewrite the matrix $J_{\text{INIS}} - (\gamma+1)\tilde{J}_{\text{INIS}}$ as the following product of block matrices:

$$\begin{aligned} &J_{\text{INIS}} - (\gamma+1)\,\tilde{J}_{\text{INIS}} = \\ &\begin{bmatrix} \mathbb{1}_{n_{\text{y}}} & \mathbb{0} & \mathbb{0} & \mathbb{0} \\ \mathbb{0} & \tilde{M} & \mathbb{0} & \mathbb{0} \\ \mathbb{0} & \mathbb{0} & -\gamma\,\mathbb{1}_{n_{\text{h}}} & \mathbb{0} \\ \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{1}_{n_{\text{D}}} \end{bmatrix} \begin{bmatrix} H - (\gamma+1)\,\tilde{H} & \tilde{g}_y^\top & h_y^\top & \mathbb{0} \\ \tilde{g}_y & \mathbb{0} & \mathbb{0} & \mathbb{0} \\ h_y & \mathbb{0} & \mathbb{0} & \mathbb{0} \\ s_y & \mathbb{0} & \mathbb{0} & \mathbb{1}_{n_{\text{w}}} \otimes \tilde{M} \end{bmatrix} \\ &\qquad \times \begin{bmatrix} \mathbb{1}_{n_{\text{y}}} & \mathbb{0} & \mathbb{0} & \mathbb{0} \\ \mathbb{0} & \tilde{M} & \mathbb{0} & \mathbb{0} \\ \mathbb{0} & \mathbb{0} & -\gamma\,\mathbb{1}_{n_{\text{h}}} & \mathbb{0} \\ \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{1}_{n_{\text{D}}} \end{bmatrix}^\top, \end{aligned} \tag{19}$$

where the matrix $\tilde{M} = g_z - (\gamma+1)\,M$ is defined, such that $\tilde{M}\tilde{g}_y = g_y - (\gamma+1)\,M\,\tilde{g}_y$. The determinant of the product of matrices in Eq. (19) can be rewritten as

$$\begin{aligned} \det\left(J_{\text{INIS}} - (\gamma+1)\tilde{J}_{\text{INIS}}\right) &= \det\left(\begin{bmatrix} \mathbb{1}_{n_{\text{y}}} & \mathbb{0} & \mathbb{0} & \mathbb{0} \\ \mathbb{0} & \tilde{M} & \mathbb{0} & \mathbb{0} \\ \mathbb{0} & \mathbb{0} & -\gamma\,\mathbb{1}_{n_{\text{h}}} & \mathbb{0} \\ \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{1}_{n_{\text{D}}} \end{bmatrix}\right)^2 \\ &\quad \times \det\left(\begin{bmatrix} H - (\gamma+1)\,\tilde{H} & \tilde{g}_y^\top & h_y^\top & \mathbb{0} \\ \tilde{g}_y & \mathbb{0} & \mathbb{0} & \mathbb{0} \\ h_y & \mathbb{0} & \mathbb{0} & \mathbb{0} \\ s_y & \mathbb{0} & \mathbb{0} & \mathbb{1}_{n_{\text{w}}} \otimes \tilde{M} \end{bmatrix}\right) \\ &= (-\gamma)^{2\,n_{\text{h}}} \det\left(\tilde{M}\right)^{2+n_{\text{w}}} \det\left(\begin{bmatrix} H - (\gamma+1)\,\tilde{H} & \tilde{g}_y^\top & h_y^\top \\ \tilde{g}_y & \mathbb{0} & \mathbb{0} \\ h_y & \mathbb{0} & \mathbb{0} \end{bmatrix}\right). \end{aligned} \tag{20}$$

Note that the Jacobian approximation M is invertible such that the determinant $\det(\tilde{M})$ is zero if and only if $\det\left(M^{-1} g_z - (\gamma+1)\,\mathbb{1}_{n_z}\right) = 0$ holds. It follows that $\det\left(J_{\text{INIS}} - (\gamma+1)\tilde{J}_{\text{INIS}}\right) = 0$ holds only for the values of γ that fulfill:

$$\gamma = 0, \qquad \text{or} \tag{21a}$$

$$\det\left(M^{-1} g_z - (\gamma+1)\,\mathbb{1}_{n_z}\right) = 0, \qquad \text{or} \tag{21b}$$

$$\det\left(\begin{bmatrix} H - (\gamma+1)\,\tilde{H} & \tilde{g}_y^\top & h_y^\top \\ \tilde{g}_y & \mathbb{0} & \mathbb{0} \\ h_y & \mathbb{0} & \mathbb{0} \end{bmatrix}\right) = 0. \tag{21c}$$

Note that Eq. (21b) is satisfied exactly for the eigenvalues $\gamma \in \sigma\left(M^{-1}g_z - \mathbb{1}_{n_z}\right)$ with an algebraic multiplicity $(n_w + 2)$ as can be observed directly in Eq. (20). It can be verified that the values for γ satisfying Eq. (21c) are given by:

$$\det\left(Z^\top\left(H - (\gamma+1)\,\tilde{H}\right)Z\right) = \det\left(H_Z - (\gamma+1)\,\tilde{H}_Z\right) = 0, \tag{22}$$

where $Z \in \mathbb{R}^{n_y \times \tilde{n}_z}$ denotes a basis for the null space of the complete constraint Jacobian c_y. The last equality in (22) is satisfied only for the eigenvalues $\gamma \in \sigma\left(\tilde{H}_Z^{-1}H_Z - \mathbb{1}_{\tilde{n}_z}\right)$. Note that this, for example, corresponds to an additional eigenvalue $\gamma = 0$ in case of an exact Hessian matrix $\tilde{H} = H$. ◻

Based on the latter results regarding the eigenspectrum of the iteration matrix, we now formally state the local contraction theorem for the proposed INIS method.

COROLLARY 8 (Local INIS-type contraction). *The local rate of convergence for the INIS-type optimization algorithm is defined by*

$$\kappa^\star_{\text{INIS}} = \rho\left(\tilde{J}_{\text{INIS}}^{-1}J_{\text{INIS}} - \mathbb{1}_{n_{\text{INIS}}}\right) = \max\left(\kappa^\star_{\text{F}},\, \rho(\tilde{H}_Z^{-1}H_Z - \mathbb{1}_{\tilde{n}_z})\right),$$

where $\kappa^\star_{\text{F}} = \rho(M^{-1}g_z - \mathbb{1}_{n_z})$ is defined for the Newton-type method on the forward problem in (3). It follows that local contraction for the forward problem, i.e. $\kappa^\star_{\text{F}} < 1$, is necessary for local convergence of the INIS-type algorithm. Under the condition $\rho(\tilde{H}_Z^{-1}H_Z - \mathbb{1}_{\tilde{n}_z}) \leq \kappa^\star_{\text{F}}$ on the quality of the Hessian approximation, e.g. $\rho(\tilde{H}_Z^{-1}H_Z - \mathbb{1}_{\tilde{n}_z}) = 0$ in case of an exact Hessian, local contraction for the forward problem is additionally sufficient since the asymptotic rate of convergence satisfies $\kappa^\star_{\text{INIS}} = \kappa^\star_{\text{F}}$.

3.3. Numerical Results. Let us first revisit the motivating QP example from Section 2.4, where the asymptotic contraction rate for the Newton-type method on the forward problem reads $\kappa^\star_{\text{F}} = 0.48 < 1$. In contrast, the solution was found to be asymptotically unstable since $\kappa^\star_{\text{IN}} \approx 1.625 > 1$ for the IN method based on the same Jacobian approximation. Let us now consider the proposed INIS algorithm based on the solution of the linear system (14) for the same QP example using the exact Hessian $\tilde{H} = H$. We compute the corresponding contraction rate at the solution

$$\kappa^\star_{\text{INIS}} = \rho(\tilde{J}_{\text{INIS}}^{-1}J_{\text{INIS}} - \mathbb{1}_{n_{\text{INIS}}}) = 0.48 < 1,$$

where J_{INIS} denotes the exact Jacobian of the augmented KKT system in (16). Therefore, the INIS scheme indeed exhibits a linear local convergence with the same asymptotic rate as the forward problem, i.e. $\kappa^\star_{\text{INIS}} = 0.48 = \kappa^\star_{\text{F}}$. This result is consistent with Theorem 7 and is illustrated in Figure 1.

In addition, let us introduce a simple example of an NLP (1) based on the QP formulation above, where again $n_h = 0$. For this purpose, let us take a quadratic objective $f(y) = \frac{1}{2}y^\top H y + h^\top y$ where H is defined in Eq. (12), the gradient vector $h = \begin{bmatrix}0.1 & 0 & 0 & 0\end{bmatrix}^\top$ and the nonlinear constraint function reads

$$g(y) = \begin{bmatrix} A_1 & A_2 \end{bmatrix} y + 0.1 \begin{bmatrix} y_1^3 \\ y_2 y_4 \end{bmatrix}, \tag{23}$$

where also the matrices A_1 and A_2 are adopted from Eq. (12). Figure 2 then illustrates the convergence results for the IN and INIS schemes from Algorithm 1 and 2 on this NLP example. It can be observed that the local contraction rate for INIS corresponds

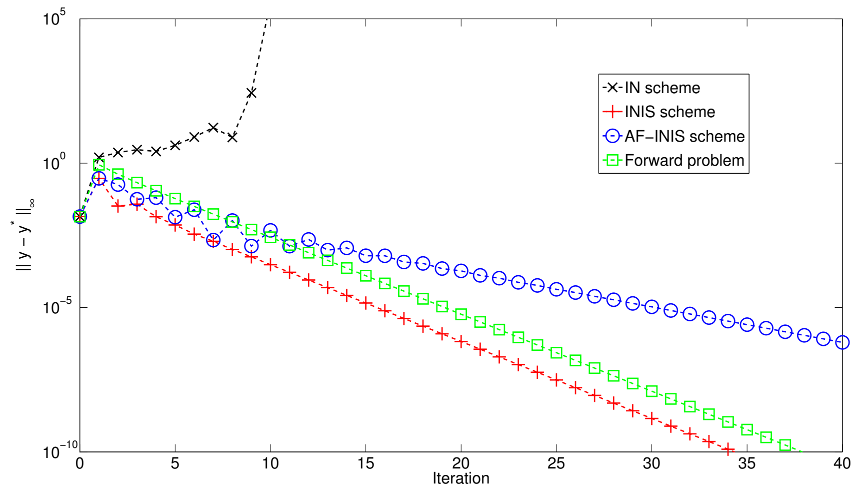

FIG. 2. *Illustration of the divergence of the Inexact Newton (IN) and the convergence of the Inexact Newton with Iterated Sensitivities (INIS) scheme for the NLP in Eq. (23). In addition, the rate of convergence for INIS can be observed to be the same as for the forward problem while the adjoint-free (AF-INIS) implementation has a different contraction rate for this NLP example.*

to that for the forward problem, while the standard IN implementation locally diverges for this particular example. More specifically, the asymptotic contraction rates at the NLP solution can be computed to be

$$\kappa_{\mathrm{F}}^{\star} = \kappa_{\mathrm{INIS}}^{\star} \approx 0.541 < 1 < 1.441 \approx \kappa_{\mathrm{IN}}^{\star}.$$

4. Adjoint-Free INIS-Type Optimization.

Algorithm 2 presented an INIS-type Newton method to solve the augmented KKT system in Eq. (13), based on adjoint sensitivity propagation to evaluate the gradient of the Lagrangian $\nabla_y \mathcal{L}(y, \lambda) = \nabla_y f(y) + \nabla_y g(y)\mu + \nabla_y h(y)\nu$. Unlike the standard IN method in Algorithm 1, for which adjoint sensitivity propagation is necessary for convergence as discussed in [9, 22], the proposed INIS algorithm allows for deploying an adjoint-free implementation as presented in this section. For this purpose, in order to motivate the use of such an adjoint-free INIS (AF-INIS) scheme, we assume the following

- A multiplier-free Hessian approximation $\tilde{H}(y) \approx H(y, \lambda) := \nabla_y^2 \mathcal{L}(y, \lambda)$ can be used for the NLP in (1), e.g., based on the Generalized Gauss-Newton method [8, 37]. It can be desirable to use a multiplier-free algorithm, which therefore does not require a good initialization of the multiplier values.
- The constraint function $g(\cdot)$ of the forward problem consists of a sequence of nontrivial operations, resulting in a Jacobian g_z with a block banded structure. For example, in the case of direct optimal control [10], these constraints typically correspond to the numerical simulation of the system dynamics. Especially for implicit integration schemes, the computation of adjoint derivatives typically results either in relatively high storage requirements of the forward variables or in an increased computational cost [45].
- Unlike the equations of the forward problem, the constraint function $h(\cdot)$ allows a relatively cheap evaluation of forward and adjoint derivatives.

The above assumptions are often satisfied for dynamic optimization problems, as discussed further in Section 5. Even though any derivative in a Newton-type method could be evaluated either forward or backward, note that there is a clear motivation to avoid the use of adjoint differentiation specifically for the function $g(\cdot)$. However, we will show, including a counterexample, that such an adjoint-free INIS method generally cannot preserve the same asymptotic contraction rate for NLPs.

Algorithm 3 One iteration of an Adjoint-Free Inexact Newton with Iterated Sensitivities (AF-INIS) optimization method.

Input: Current values $\bar{y} = (\bar{z}, \bar{w})$, $\bar{D}$ and approximations M, $\tilde{H}(\bar{y})$.

1: After eliminating the variables Δz, $\Delta \mu$ in (25), solve the resulting system:
$$\begin{bmatrix} \tilde{Z}^\top \tilde{H} \tilde{Z} & \tilde{Z}^\top h_y^\top \\ h_y \tilde{Z} & \mathbb{0} \end{bmatrix} \begin{bmatrix} \Delta w \\ \bar{\nu}^+ \end{bmatrix} = - \begin{bmatrix} \tilde{Z}^\top \nabla_y f(\bar{y}) \\ h(\bar{y}) \end{bmatrix} - \begin{bmatrix} \tilde{Z}^\top \tilde{H} \\ h_y \end{bmatrix} \begin{bmatrix} -M^{-1} g(\bar{y}) \\ \mathbb{0} \end{bmatrix},$$
where $\tilde{Z}^\top := \left[-\bar{D}^\top, \ \mathbb{1}_{n_\mathrm{w}} \right]$.

2: Based on Δw, the corresponding value for Δz is found:
$\Delta z = -M^{-1} g(\bar{y}) - \bar{D} \Delta w$.

3: Independently, the sensitivity matrix is updated in each iteration:
$\Delta D = -M^{-1} (g_z \bar{D} - g_w)$.

Output: New values $\bar{y}^+ = \bar{y} + \Delta y$ and $\bar{D}^+ = \bar{D} + \Delta D$.

4.1. Implementation. Algorithm 3 presents the adjoint-free variant of the INIS optimization method from Algorithm 2. It corresponds to solving the following approximate variant of the augmented KKT system in Eq. (13):

$$\mathcal{F}_{\mathrm{AF}}(y, \lambda, D) = \begin{bmatrix} \nabla_y f(y) + \begin{pmatrix} g_z^\top \\ D^\top g_z^\top \end{pmatrix} \mu + \nabla_y h(y) \nu \\ c(y) \\ \mathrm{vec}(g_z D - g_w) \end{bmatrix} = 0. \tag{24}$$

The following proposition formalizes the connection between this augmented system of equations and the original NLP in Eq. (1).

PROPOSITION 9. *A solution $(y^\star, \lambda^\star, D^\star)$ to the alternative augmented system in Eq. (24), corresponds to a regular KKT point $(y^\star, \lambda^\star)$ for the NLP in Eq. (1).*

Proof. The third equation in both augmented KKT systems from Eq. (13) and Eq. (24) at the solution $(y^\star, \lambda^\star, D^\star)$ reads as $g_z D^\star - g_w = 0$ such that $D^\star = g_z^{-1} g_w$ holds. The following equality therefore holds at the solution:

$$\nabla_y \mathcal{L}(y^\star, \lambda^\star) = \nabla_y f(y^\star) + \nabla_y c(y^\star) \lambda^\star = \nabla_y f(y^\star) + \begin{pmatrix} g_z^\top \\ (g_z D^\star)^\top \end{pmatrix} \mu^\star + \nabla_y h(y^\star) \nu^\star.$$

It follows that a solution of the adjoint-free augmented system (24) also forms a solution to the adjoint-based augmented system (13) and therefore is a regular KKT point for the NLP in Eq. (1) based on the result in Proposition 6. ∎

The adjoint-free inexact Newton method with iterated sensitivities (AF-INIS) then uses the same approximate Jacobian matrix $\tilde{J}_{\mathrm{INIS}}(\bar{y}, \bar{\lambda}, \bar{D})$ from Eq. (14) to solve the augmented set of equations in (24). At each iteration, the corresponding linear system reads as

$$
\underbrace{\begin{bmatrix} \tilde{H} & \begin{pmatrix} M^\top & h_z^\top \\ \bar{D}^\top M^\top & h_w^\top \end{pmatrix} & \mathbb{0} \\ \begin{pmatrix} M & M\bar{D} \\ h_z & h_w \end{pmatrix} & \mathbb{0} & \mathbb{0} \\ \mathbb{0} & \mathbb{0} & \mathbb{1}_{n_\mathrm{w}} \otimes M \end{bmatrix}}_{=\tilde{J}_{\mathrm{INIS}}(\bar{y},\bar{\lambda},\bar{D})} \begin{bmatrix} \Delta z \\ \Delta w \\ \Delta \mu \\ \Delta \nu \\ \mathrm{vec}(\Delta D) \end{bmatrix} \tag{25}
$$
$$
= - \underbrace{\begin{bmatrix} \nabla_y f(\bar{y}) + \begin{pmatrix} g_z^\top \\ \bar{D}^\top g_z^\top \end{pmatrix} \bar{\mu} + \nabla_y h(\bar{y}) \bar{\nu} \\ c(\bar{y}) \\ \mathrm{vec}(g_z \bar{D} - g_w) \end{bmatrix}}_{=\mathcal{F}_{\mathrm{AF}}(\bar{y},\bar{\lambda},\bar{D})}.
$$

Using this augmented linear system, the steps Δz, Δw and ΔD can be computed without evaluating adjoint derivatives for the function $g(\cdot)$ in Algorithm 3. The evaluation of these adjoint variables can be avoided because the following term vanishes when multiplying the first equation in the right-hand side of the latter system (25) by $\tilde{Z}^\top := \begin{bmatrix} -\bar{D}^\top & \mathbb{1}_{n_\mathrm{w}} \end{bmatrix}$:

$$
\begin{bmatrix} -\bar{D}^\top & \mathbb{1}_{n_\mathrm{w}} \end{bmatrix} \begin{pmatrix} g_z^\top \\ \bar{D}^\top g_z^\top \end{pmatrix} = -\bar{D}^\top g_z^\top + \bar{D}^\top g_z^\top = \mathbb{0},
$$

resulting in an adjoint-free and multiplier-free computation in Algorithm 3. Note that the multipliers ν are also not needed, depending on how the linear system is solved in step 1 of the algorithm. Because we assumed the Hessian approximation $\tilde{H}(\bar{y})$ in this case to be independent of the current multiplier values, we can completely omit the computation of the update $\Delta\lambda$.

4.2. Local Convergence Results. Proposition 9 states that, if it converges, the adjoint-free implementation of the INIS method in Algorithm 3 converges to a local minimizer for the NLP in Eq. (1), and this unlike standard adjoint-free inexact Newton methods as discussed in [9, 22]. Even though we will show that the result in Theorem 7 does not necessarily hold for the AF-INIS scheme applied to general NLPs, the following theorem extends this local contraction result for quadratic programming (QP) problems. Let us introduce the exact Jacobian of the adjoint-free augmented KKT system in Eq. (24):

$$
J_{\mathrm{AF}}(y, \lambda, D) = \begin{bmatrix} f_{yy} + \tilde{g}_{yy} + h_{yy} & \begin{pmatrix} g_z^\top & h_z^\top \\ D^\top g_z^\top & h_w^\top \end{pmatrix} & \tilde{g}_D \\ \begin{pmatrix} g_z & g_w \\ h_z & h_w \end{pmatrix} & \mathbb{0} & \mathbb{0} \\ s_y & \mathbb{0} & \mathbb{1}_{n_\mathrm{w}} \otimes g_z \end{bmatrix}, \tag{26}
$$

where the matrices $f_{yy} := \nabla_y^2 f(y)$, $\tilde{g}_{yy} := \frac{\partial}{\partial y} \begin{pmatrix} g_z^\top \mu \\ D^\top g_z^\top \mu \end{pmatrix}$, $h_{yy} := \sum_{i=1}^{n_\mathrm{h}} \nabla_y^2 h_i(y) \nu_i$ and $\tilde{g}_D := \begin{pmatrix} \mathbb{0} \\ \mathbb{1}_{n_\mathrm{w}} \otimes \mu^\top g_z \end{pmatrix}$ are defined and $s_y := \frac{\partial}{\partial y} \mathrm{vec}(g_z D - g_w)$ similar to Eq. (16). For this local convergence result, we consider a QP of the form in Eq. (1):

$$
\begin{aligned}
&\text{(27a)} && \min_{z,w} && \frac{1}{2}y^\top H y + h^\top y \\
&\text{(27b)} && \text{s.t.} && A_1 z + A_2 w + a = 0, \\
&\text{(27c)} && && B_1 z + B_2 w + b = 0,
\end{aligned}
$$

where the matrix A_1 is assumed to be invertible and we have an invertible Jacobian approximation $M \approx A_1$ available.

THEOREM 10. *For the adjoint-free augmented linear system (25) corresponding to the QP in Eq. (27), the eigenspectrum of the AF-INIS iteration matrix reads*

$$\sigma\left(\tilde{J}_{\mathrm{INIS}}^{-1} J_{\mathrm{AF}} - \mathbb{1}_{n_{\mathrm{INIS}}}\right) = \{0\} \cup \sigma\left(M^{-1}A_1 - \mathbb{1}_{n_z}\right) \cup \sigma\left(\tilde{H}_Z^{-1} H_Z - \mathbb{1}_{\tilde{n}_Z}\right), \tag{28}$$

at the solution $(y^\star, \lambda^\star, D^\star)$. The exact Jacobian $J_{\mathrm{AF}}(y, \lambda, D)$ is defined by Eq. (26) for which $s_y = \mathbb{0}$, $\tilde{g}_{yy} = \mathbb{0}$, $h_{yy} = \mathbb{0}$ and $f_{yy} = H$ in case of a QP formulation. Similar to Theorem 7, $\tilde{n}_{\mathrm{Z}} = n_{\mathrm{w}} - n_{\mathrm{h}}$ and $Z \in \mathbb{R}^{n_{\mathrm{y}} \times \tilde{n}_{\mathrm{Z}}}$ denotes a basis for the null space of the constraint Jacobian $\begin{bmatrix} A \\ B \end{bmatrix}$, and $H_Z := Z^\top H Z \in \mathbb{R}^{\tilde{n}_{\mathrm{Z}} \times \tilde{n}_{\mathrm{Z}}}$ and $\tilde{H}_Z := Z^\top \tilde{H} Z \in \mathbb{R}^{\tilde{n}_{\mathrm{Z}} \times \tilde{n}_{\mathrm{Z}}}$. The local rate of convergence for the adjoint-free INIS scheme on the QP formulation in (27) is defined by

$$\kappa_{\mathrm{AF}}^\star = \rho\left(\tilde{J}_{\mathrm{INIS}}^{-1} J_{\mathrm{AF}} - \mathbb{1}_{n_{\mathrm{INIS}}}\right) = \max\left(\kappa_{\mathrm{F}}^\star,\, \rho(\tilde{H}_Z^{-1} H_Z - \mathbb{1}_{\tilde{n}_{\mathrm{Z}}})\right).$$

Proof. At the solution of the adjoint-free augmented KKT system in Eq. (24) for the QP formulation in (27), we know that $D^\star = A_1^{-1} A_2$ and let us use the notation $A = \begin{bmatrix} A_1 & A_2 \end{bmatrix}$ and $\tilde{A} = A_1^{-1} A$. The eigenvalues γ of the iteration matrix $\tilde{J}_{\mathrm{INIS}}^{-1} J_{\mathrm{AF}} - \mathbb{1}_{n_{\mathrm{INIS}}}$ are given by the expression $\det\left(J_{\mathrm{AF}} - (\gamma+1)\tilde{J}_{\mathrm{INIS}}\right) = 0$, based on the exact and inexact adjoint-free augmented Jacobian matrices

$$J_{\mathrm{AF}} = \begin{bmatrix} H & A^\top & B^\top & \tilde{g}_D \\ A & \mathbb{0} & \mathbb{0} & \mathbb{0} \\ B & \mathbb{0} & \mathbb{0} & \mathbb{0} \\ \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{1}_{n_{\mathrm{w}}} \otimes A_1 \end{bmatrix}, \quad \tilde{J}_{\mathrm{INIS}} = \begin{bmatrix} \tilde{H} & \tilde{A}^\top M^\top & B^\top & \mathbb{0} \\ M\tilde{A} & \mathbb{0} & \mathbb{0} & \mathbb{0} \\ B & \mathbb{0} & \mathbb{0} & \mathbb{0} \\ \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{1}_{n_{\mathrm{w}}} \otimes M \end{bmatrix}, \tag{29}$$

where $\tilde{g}_D = \begin{pmatrix} \mathbb{0} \\ \mathbb{1}_{n_{\mathrm{w}}} \otimes {\mu^\star}^\top A_1 \end{pmatrix}$ is defined at the solution point $(y^\star, \lambda^\star, D^\star)$. We can rewrite $J_{\mathrm{AF}} - (\gamma+1)\tilde{J}_{\mathrm{INIS}}$ as the following product of block matrices:

$$
\begin{aligned}
&J_{\mathrm{AF}} - (\gamma+1)\,\tilde{J}_{\mathrm{INIS}} = \\
&\begin{bmatrix} \mathbb{1}_{n_{\mathrm{y}}} & \mathbb{0} & \mathbb{0} & \mathbb{0} \\ \mathbb{0} & \tilde{M} & \mathbb{0} & \mathbb{0} \\ \mathbb{0} & \mathbb{0} & -\gamma\,\mathbb{1}_{n_{\mathrm{h}}} & \mathbb{0} \\ \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{1}_{n_{\mathrm{D}}} \end{bmatrix}
\begin{bmatrix} H - (\gamma+1)\,\tilde{H} & \tilde{A}^\top & B^\top & \tilde{g}_D \\ \tilde{A} & \mathbb{0} & \mathbb{0} & \mathbb{0} \\ B & \mathbb{0} & \mathbb{0} & \mathbb{0} \\ \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{1}_{n_{\mathrm{w}}} \otimes \tilde{M} \end{bmatrix} \\
&\qquad\times \begin{bmatrix} \mathbb{1}_{n_{\mathrm{y}}} & \mathbb{0} & \mathbb{0} & \mathbb{0} \\ \mathbb{0} & \tilde{M} & \mathbb{0} & \mathbb{0} \\ \mathbb{0} & \mathbb{0} & -\gamma\,\mathbb{1}_{n_{\mathrm{h}}} & \mathbb{0} \\ \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{1}_{n_{\mathrm{D}}} \end{bmatrix}^\top,
\end{aligned} \tag{30}
$$

where $\tilde{M} = A_1 - (\gamma+1)\, M$ is defined. The proof continues in the same way as for Theorem 7, since the determinant of this matrix can be written as

$$\begin{aligned}&\det\left(J_{\mathrm{AF}} - (\gamma+1)\tilde{J}_{\mathrm{INIS}}\right)\\ &= (-\gamma)^{2\, n_{\mathrm{h}}} \det\left(\tilde{M}\right)^{2+n_{\mathrm{w}}} \det\left(\begin{bmatrix} H - (\gamma+1)\,\tilde{H} & \tilde{A}^\top & B^\top \\ \tilde{A} & \mathbb{0} & \mathbb{0} \\ B & \mathbb{0} & \mathbb{0}\end{bmatrix}\right).\end{aligned} \tag{31}$$ □

4.3. Remark on AF-INIS for NLPs. When applying the adjoint-free INIS scheme from Algorithm 3 to the NLP formulation in Eq. (1), the augmented system introduces off-diagonal blocks for the Jacobian matrix as defined in Eq. (26). Therefore, the local contraction result in Theorem 10 cannot be directly extended to the general NLP case, even though the practical convergence of AF-INIS can typically be expected to be similar for relatively mild nonlinearities in the problem formulation. Note that Figure 1 already illustrated the local convergence of the AF-INIS scheme on the QP in (12), for which the following holds

$$\kappa^\star_{\mathrm{F}} = \kappa^\star_{\mathrm{INIS}} = \kappa^\star_{\mathrm{AF}} = 0.48 < 1 < 1.625 \approx \kappa^\star_{\mathrm{IN}}.$$

Note that Section 3.3 included a counterexample to the conjecture that Theorem 10 could hold for general NLPs. It can be observed from Figure 2 that the local convergence rate of AF-INIS is different from the adjoint based INIS scheme, i.e.

$$\kappa^\star_{\mathrm{F}} = \kappa^\star_{\mathrm{INIS}} \approx 0.541 < \kappa^\star_{\mathrm{AF}} \approx 0.753 < 1 < 1.441 \approx \kappa^\star_{\mathrm{IN}},$$

even though it still outperforms the standard inexact Newton (IN) method.

5. Applications and Numerical Results. This section motivates the practical applicability of the INIS-type optimization method, either with or without adjoint computation respectively in Algorithm 2 or 3. For this purpose, let us introduce simultaneous direct optimal control methods for the popular class of dynamic optimization problems which typically have the form in Eq. (1), where the functions $f(\cdot)$, $g(\cdot)$ and $h(\cdot)$ are twice continuously differentiable and the Jacobian matrix g_z is invertible. Similar to before, this discussion omits the presence of inequality constraints even though the above results on local Newton-type convergence can be extended. This will be illustrated based on numerical results for the chain mass example [49].

5.1. Direct Optimal Control. In direct optimal control [10], one applies a *first-discretize-then-optimize* type of approach where one first discretizes the continuous time Optimal Control Problem (OCP) such that one can subsequently solve an NLP of the form in Eq. (1). In case of *direct collocation* [5], such a discrete-time OCP problem can for example read as:

$$\min_{X,\,U,\,K} \quad \sum_{i=0}^{N-1} l_i(x_i,u_i) + l_N(x_N) \tag{32a}$$

$$\text{s.t.} \quad 0 = c_i(x_i,u_i,K_i), \qquad i = 0,\dots,N-1, \tag{32b}$$

$$0 = x_0 - \hat{x}_0, \tag{32c}$$

$$0 = x_i + B_i\, K_i - x_{i+1}, \quad i = 0,\dots,N-1, \tag{32d}$$

with differential states $x_i \in \mathbb{R}^{n_x}$, control inputs $u_i \in \mathbb{R}^{n_u}$ and collocation variables $K_i \in \mathbb{R}^{qN_\mathrm{s}n_x}$, in which q is the number of collocation nodes and N_s the amount of integration steps. In addition, the state $X = [x_0^\top, \dots, x_N^\top]^\top$ and control trajectory $U = [u_0^\top, \dots, u_{N-1}^\top]^\top$ and the trajectory of collocation variables $K = [K_0^\top, \dots, K_{N-1}^\top]^\top$ are defined. The function $l_i(\cdot)$ denotes the stage cost and $c_i : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_K} \to \mathbb{R}^{n_K}$ defines the collocation polynomial on each interval $i = 0, \dots, N-1$, where N denotes the number of intervals in the control horizon.

When comparing this OCP to the general NLP formulation in Eq. (1), similar to the detailed discussion in [44], one can relate the variables $z = [K_0^\top, \dots, K_{N-1}^\top]^\top$ and $w = [x_0^\top, u_0^\top, \dots, u_{N-1}^\top, x_N^\top]^\top$. Given the state and control values in w, the nonlinear collocation equations (32b) form the function $g(\cdot)$ that defines all variables in z as required for the problem formulation in (1). The additional equality constraints from (1c) then correspond to the initial value constraint in (32c) and the continuity constraints in (32d). Based on Remark 5, note that the Newton-type local convergence results in this article still hold for inequality constrained optimization problems under certain regularity conditions in a neighborhood of the local minimizer. This will also be illustrated numerically in Subsection 5.3. The Newton-type optimization algorithms in this article can rely on an efficient approximation of the invertible Jacobian $M_i \approx \frac{\partial c_i}{\partial K_i}$ as discussed in [3, 12, 13, 26, 42] for collocation methods. These collocation variables could be numerically eliminated in each iteration, based on the constraints in Eq. (32b), resulting in a multiple shooting type method as discussed in [43, 44]. It is important to note that the sensitivity matrix variable in INIS-type optimization has a block-diagonal structure here because of the stage-wise definition of the collocation equations in (32b), i.e., $D_i \in \mathbb{R}^{n_K \times (n_x+n_u)}$ can be defined for $i = 0, \dots, N-1$. In addition, the conditions in Section 4 are satisfied, for example, in case of a (nonlinear) least squares type objective in (32a), for which a Gauss-Newton Hessian approximation can be used. The constraints in (32c) and (32d) are linear in this OCP formulation, while the nonlinear collocation equations in (32b) can correspond to a sequence of integration steps for which adjoint differentiation could be avoided in the AF-INIS optimization algorithm.

Similar to the formulation in [44], we can write the collocation equations in (32b) for one interval $i = 0, \dots, N-1$:

$$c_i(x_i, u_i, K_i) = \begin{bmatrix} c_{i,1}(x_{i,0}, u_i, K_{i,1}) \\ \vdots \\ c_{i,N_\mathrm{s}}(x_{i,N_\mathrm{s}-1}, u_i, K_{i,N_\mathrm{s}}) \end{bmatrix} = 0, \tag{33}$$

for N_s integration steps of a q-stage collocation method. Note that $x_{i,j} \in \mathbb{R}^{n_x}$ for $j = 1, \dots, N_\mathrm{s}$ denote the intermediate state values, $x_{i,0} = x_i$ and $K_{i,j} \in \mathbb{R}^{qn_x}$ is defined, such that $K_i = [K_{i,1}^\top, \dots, K_{i,N_\mathrm{s}}^\top]^\top$. This sequential simulation structure in Eq. (33) results in a constraint Jacobian that is block banded, as well as its approximation $M_i \approx \frac{\partial c_i}{\partial K_i}$. As discussed in detail in [44], this particular structure can be exploited by performing a forward and a backward propagation sweep, respectively for the condensing and the expansion step of the adjoint-based schemes in Algorithm 1 and 2. In case of an adjoint-free implementation, based on Algorithm 3, this procedure instead reduces to a forward propagation sweep.

5.2. ACADO Code Generation Tool. An open-source implementation of the INIS-type optimization algorithm for the direct collocation based OCP formulation in Eq. (32) is part of the `ACADO` Toolkit [32]. Presented as lifted collocation integrators

Table 1
Average timing results per Gauss-Newton based SQP iteration on the chain mass problem using direct collocation ($N_s = 3, q = 4$), including different numbers of masses n_m and states n_x.

n_m	n_x	Gauss-Newton	IN	INIS	AF-INIS
3	12	5.33 ms	2.40 ms	2.19 ms	1.95 ms
4	18	14.79 ms	5.43 ms	4.76 ms	4.29 ms
5	24	34.04 ms	10.71 ms	9.39 ms	7.96 ms
6	30	62.08 ms	18.73 ms	14.88 ms	12.71 ms
7	36	106.57 ms	36.09 ms	21.93 ms	20.06 ms

in [44], the methods have more specifically been implemented as part of the `ACADO` code generation tool. This package can be used to obtain real-time feasible code for dynamic optimization on embedded control hardware. In particular, it pursues the export of efficient C-code based on the Real-Time Iteration (RTI) scheme for Nonlinear MPC (NMPC) [21, 33]. This online algorithm is based on Sequential Quadratic Programming (SQP) to solve the nonlinear optimization problem within direct multiple shooting [10]. Regarding the INIS-type implementation following Algorithm 2 and 3, tailored Jacobian approximations are used for collocation methods, based on either Simplified or Single Newton-type iterations as presented in [42]. As discussed earlier in Section 4.1, a multiplier-free Hessian approximation such as in the Generalized Gauss-Newton method [8] is used for the adjoint-free variant (AF-INIS). The standard INIS algorithm can rely on any approximation technique, including an exact Hessian based approach [37]. Similar to the implementation described in Algorithm 2 and 3, condensing and expansion techniques are used to obtain multiple shooting structured subproblems in each iteration of the SQP algorithm [44]. Tailored convex solvers such as `qpOASES` [23], `qpDUNES` [24] and `HPMPC` [25] can be used to solve these subproblems, especially in the presence of inequality constraints.

5.3. Numerical Results: Chain of Masses. We consider the chain mass optimal control problem from [44, 49]. The objective is to return a chain of n_m masses connected with springs to its steady state, starting from a perturbed initial configuration. The mass at one end is fixed, while the control input $u \in \mathbb{R}^3$ to the system is the direct force applied to the mass at the other end of the chain. The state of each free mass $x^j := \begin{bmatrix} p^{j\top} & v^{j\top} \end{bmatrix}^\top \in \mathbb{R}^6$ for $j = 1, \dots, n_m - 1$ consists in its position and velocity, such that the dynamic system can be described by the concatenated state vector $x(t) \in \mathbb{R}^{6(n_m-1)}$. More details on the resulting model equations can be found in [49]. The OCP problem formulation is adopted from [44]. In addition to the constraints in Eq. (32), this OCP includes simple bounds on the control inputs and the path constraint that the chain should not hit a wall placed close to the equilibrium state. The `ACADO` code generation tool is used to generate an SQP type algorithm to solve the resulting inequality constrained optimization problem. Since the stage cost in the objective (32a) represents minimizing the control effort in the least squares sense, a Gauss-Newton based Hessian approximation will be used in this numerical case study. In addition, each SQP subproblem is solved using the parametric active-set solver `qpOASES` [23] in combination with a condensing technique to numerically eliminate the state variables [10].

Table 1 presents average timing results per Gauss-Newton based SQP iteration of the automatic generated solver using the ACADO toolkit, and this for different numbers of masses n_{m} [2]. Note that the IN, INIS and AF-INIS schemes correspond to the proposed implementations in Algorithm 1, 2 and 3, based on the lifted collocation integrators as presented in [44]. On the other hand, the exact Gauss-Newton method in this case is based on a direct solution of the QP subproblem, corresponding to the linearized KKT conditions (6) including a Gauss-Newton Hessian approximation. The table shows that the use of inexact Jacobian approximations, tailored for collocation methods [42], can considerably reduce the computational effort over an exact implementation. More specifically, the Single Newton implementation from [27] has been used for the 4-stage Gauss collocation scheme ($q = 4$). A speedup of about factor 5 can be observed for the INIS-type scheme on this particular example. Figure 3 illustrates the convergence results for the SQP method, based on these different Newton-type optimization techniques. The figure shows a simulation result for which the inexact Newton (IN) scheme still results in local convergence, even though the contraction rate can be observed to be considerably slower than both of the variants of the proposed INIS algorithm.

Note that the Gauss-Newton based Hessian approximation does not depend on the multipliers for the equality constraints, but the convergence of both the adjoint-based IN and INIS scheme in Algorithm 1 and 2 does depend on the initialization of these Lagrange multipliers unlike the adjoint-free (AF-INIS) variant. For simplicity, these multipliers have been initialized using zero values to obtain the numerical results in this case study. This difference in convergence behavior can also be observed in Figure 3. Even though the convergence for both INIS-type variants is close to that for the Newton-type method on the forward problem of this example, the contraction result in Theorem 7 cannot generally be extended to the AF-INIS algorithm for nonlinear optimization as discussed in Section 4.3. The results for the exact Gauss-Newton method have been included mainly as a reference. It namely induces a relatively high computational cost as illustrated by Table 1, especially in case only rather low accuracy results are sufficient.

6. Conclusions.

This article presented a novel family of optimization algorithms, based on inexact Newton-type iterations with iterated sensitivities (INIS). Unlike standard inexact Newton methods, this technique is shown to preserve the local contraction properties of the forward problem, based on a specific Jacobian approximation for the corresponding equality constraints. More specifically, local convergence for the Newton-type method on the forward problem is shown to be necessary, and under mild conditions even sufficient for the asymptotic contraction of the corresponding INIS-type optimization algorithm. The article presents how this INIS algorithm can be implemented efficiently, resulting in a computational cost close to that of the standard inexact Newton implementation. In addition, an adjoint-free (AF-INIS) variant is proposed and its local convergence properties are studied. This alternative approach can be preferable whenever the algorithm can be carried out independent of the current values for the multipliers corresponding to the equality constraints. Finally, an open-source implementation of these INIS-type techniques for simultaneous direct optimal control has been presented as part of the ACADO Toolkit. Theoretical results are illustrated using toy examples of optimization problems, in addition to the benchmark case study of the optimal control for a chain of masses.

[2]All numerical simulations are carried out on a standard computer, equipped with an Intel i7-3720QM processor, using a 64-bit version of Ubuntu 14.04 and the g++ compiler version 4.8.4.

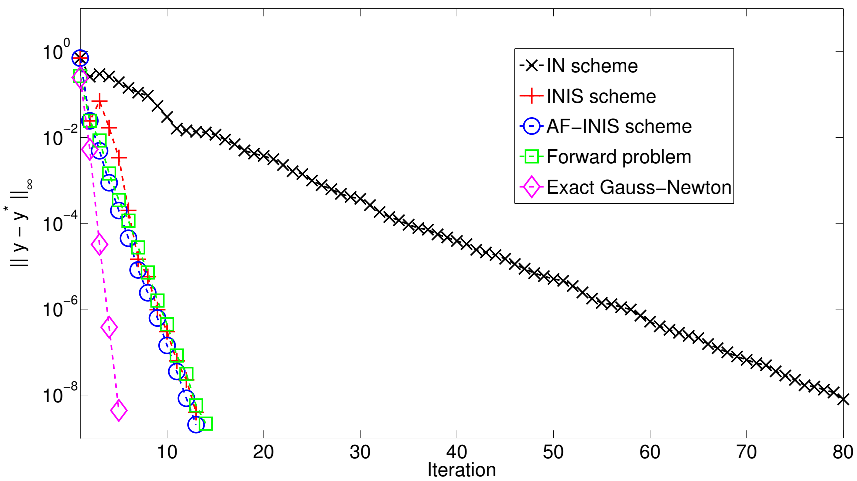

FIG. 3. *Convergence results of the Gauss-Newton based SQP method with different inexact Newton-type techniques for the chain mass optimal control problem using* $n_{\mathrm{m}} = 4$ *masses.*

REFERENCES

[1] J. Albersmeyer and M. Diehl, *The lifted Newton method and its application in optimization*, SIAM Journal on Optimization, 20 (2010), pp. 1655–1684.

[2] J. Betts, *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*, SIAM, 2nd ed., 2010.

[3] T. A. Bickart, *An efficient solution process for implicit Runge-Kutta methods*, SIAM Journal on Numerical Analysis, 14 (1977), pp. 1022–1027.

[4] L. Biegler, O. Ghattas, M. Heinkenschloss, and B. van Bloemen Waanders (eds.), *Large-Scale PDE-Constrained Optimization*, vol. 30 of Lecture Notes in Computational Science and Engineering, Springer-Verlag, 2003.

[5] L. T. Biegler, *Nonlinear Programming*, MOS-SIAM Series on Optimization, SIAM, 2010.

[6] H. Bock, *Randwertproblemmethoden zur Parameteridentifizierung in Systemen nichtlinearer Differentialgleichungen*, vol. 183 of Bonner Mathematische Schriften, Universität Bonn, Bonn, 1987.

[7] H. Bock, W. Egartner, W. Kappis, and V. Schulz, *Practical Shape Optimization for Turbine and Compressor Blades by the Use of PRSQP Methods*, Optimization and Engineering, 3 (2002), pp. 395–414.

[8] H. G. Bock, *Recent advances in parameter identification techniques for ODE*, in Numerical Treatment of Inverse Problems in Differential and Integral Equations, Birkhäuser, 1983, pp. 95–121.

[9] H. G. Bock, M. Diehl, E. A. Kostina, and J. P. Schlöder, *Constrained optimal feedback control of systems governed by large differential algebraic equations*, in Real-Time and Online PDE-Constrained Optimization, SIAM, 2007, pp. 3–22.

[10] H. G. Bock and K. J. Plitt, *A multiple shooting algorithm for direct solution of optimal control problems*, in Proceedings of the IFAC World Congress, Pergamon Press, 1984, pp. 242–247.

[11] P. T. Boggs and J. W. Tolle, *Sequential quadratic programming*, Acta Numerica, (1995), pp. 1–51.

[12] J. Butcher, *On the implementation of implicit Runge-Kutta methods*, BIT Numerical Mathematics, 16 (1976), pp. 237–240.

[13] G. Cooper and R. Vignesvaran, *Some schemes for the implementation of implicit Runge-Kutta methods*, Journal of Computational and Applied Mathematics, 45 (1993), pp. 213–225.

[14] F. E. Curtis, T. C. Johnson, D. P. Robinson, and A. Wächter, *An inexact sequential quadratic optimization algorithm for nonlinear optimization*, SIAM Journal on Optimiza-

tion, 24 (2014), pp. 1041–1074.

[15] F. E. Curtis, J. Nocedal, and A. Wächter, *A matrix-free algorithm for equality constrained optimization problems with rank-deficient Jacobians*, SIAM Journal on Optimization, 20 (2009), pp. 1224–1249.

[16] R. Dembo, S. Eisenstat, and T. Steihaug, *Inexact Newton methods*, SIAM Journal of Numerical Analysis, 19 (1982), pp. 400–408.

[17] J. E. Dennis, *On Newton-like methods*, Numerische Mathematik, 11 (1968), pp. 324–330.

[18] J. E. Dennis and J. J. Moré, *Quasi-Newton Methods, Motivation and Theory*, SIAM Review, 19 (1977), pp. 46–89.

[19] P. Deuflhard, *Newton methods for nonlinear problems: affine invariance and adaptive algorithms*, vol. 35, Springer, 2011.

[20] M. Diehl, *Lecture Notes on Numerical Optimization*, 2016. (Available online: http://cdn.syscop.de/publications/Diehl2016.pdf).

[21] M. Diehl, H. G. Bock, J. Schlöder, R. Findeisen, Z. Nagy, and F. Allgöwer, *Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations*, Journal of Process Control, 12 (2002), pp. 577–585.

[22] M. Diehl, A. Walther, H. G. Bock, and E. Kostina, *An adjoint-based SQP algorithm with quasi-Newton Jacobian updates for inequality constrained optimization*, Optimization Methods and Software, 25 (2010), pp. 531–552.

[23] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl, *qpOASES: a parametric active-set algorithm for quadratic programming*, Mathematical Programming Computation, 6 (2014), pp. 327–363.

[24] J. V. Frasch, S. Sager, and M. Diehl, *A parallel quadratic programming method for dynamic optimization problems*, Mathematical Programming Computations, 7 (2015), pp. 289–329.

[25] G. Frison, H. B. Sorensen, B. Dammann, and J. B. Jørgensen, *High-performance small-scale solvers for linear model predictive control*, in Proceedings of the European Control Conference (ECC), June 2014, pp. 128–133.

[26] S. González-Pinto, J. I. Montijano, and L. Rández, *Iterative schemes for three-stage implicit Runge-Kutta methods*, Appl. Numer. Math., 17 (1995), pp. 363–382.

[27] S. González-Pinto, S. Pérez-Rodríguez, and J. I. Montijano, *Implementation of high-order implicit Runge-Kutta methods*, Computers & Mathematics with Applications, 41 (2001), pp. 1009–1024.

[28] A. Griewank, *Evaluating Derivatives, Principles and Techniques of Algorithmic Differentiation*, no. 19 in Frontiers in Appl. Math., SIAM, Philadelphia, 2000.

[29] A. Griewank and A. Walther, *On Constrained Optimization by Adjoint based quasi-Newton Methods*, Optimization Methods and Software, 17 (2002), pp. 869–889.

[30] M. Heinkenschloss and L. Vicente, *Analysis of Inexact Trust-Region SQP Algorithms*, SIAM Journal on Optimization, 12 (2001), pp. 283–302.

[31] B. Houska and M. Diehl, *A quadratically convergent inexact SQP method for optimal control of differential algebraic equations*, Optimal Control Applications and Methods, 34 (2013), pp. 396–414.

[32] B. Houska, H. J. Ferreau, and M. Diehl, *ACADO toolkit – an open source framework for automatic control and dynamic optimization*, Optimal Control Applications and Methods, 32 (2011), pp. 298–312.

[33] B. Houska, H. J. Ferreau, and M. Diehl, *An auto-generated real-time iteration algorithm for nonlinear MPC in the microsecond range*, Automatica, 47 (2011), pp. 2279–2285.

[34] H. Jaeger and E. Sachs, *Global convergence of inexact reduced SQP methods*, Optimization Methods and Software, 7 (1997), pp. 83–110.

[35] T. C. Johnson, C. Kirches, and A. Wächter, *An active-set method for quadratic programming based on sequential hot-starts*, SIAM Journal on Optimization, 25 (2015), pp. 967–994.

[36] F. Leibfritz and E. W. Sachs, *Inexact SQP interior point methods and large scale optimal control problems*, SIAM Journal on Control and Optimization, 38 (2006), pp. 272–293.

[37] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer Series in Operations Research and Financial Engineering, Springer, 2 ed., 2006.

[38] A. Potschka, *Handling Path Constraints in a Direct Multiple Shooting Method for Optimal Control Problems*, Diplomarbeit, University of Heidelberg, 2006.

[39] A. Potschka, *A direct method for the numerical solution of optimization problems with time-periodic PDE constraints*, PhD thesis, University of Heidelberg, 2011.

[40] A. Potschka, H. Bock, S. Engell, A. Küpper, and J. Schlöder, *A Newton-Picard inexact SQP method for optimization of SMB processes*, tech. report, 2008.

[41] R. Quirynen, *Numerical Simulation Methods for Embedded Optimization*, PhD thesis, KU

Leuven and University of Freiburg, 2017.
[42] R. Quirynen, S. Gros, and M. Diehl, *Inexact Newton based lifted implicit integrators for fast nonlinear MPC*, in Proceedings of the IFAC Conference on Nonlinear Model Predictive Control (NMPC), 2015, pp. 32–38.
[43] R. Quirynen, S. Gros, and M. Diehl, *Lifted implicit integrators for direct optimal control*, in Proceedings of the IEEE Conference on Decision and Control (CDC), 2015, pp. 3212–3217.
[44] R. Quirynen, S. Gros, B. Houska, and M. Diehl, *Lifted collocation integrators for direct optimal control in ACADO toolkit*, Mathematical Programming Computation (accepted, preprint available at Optimization Online, 2016-05-5468), (2016).
[45] R. Quirynen, B. Houska, and M. Diehl, *Efficient symmetric Hessian propagation for direct optimal control*, Journal of Process Control, 50 (2017), pp. 19–28.
[46] S. Robinson, *Perturbed Kuhn-Tucker points and rates of convergence for a class of nonlinear programming algorithms*, Mathematical Programming, 7 (1974), pp. 1–16.
[47] A. Walther and L. Biegler, *Numerical experiments with an inexact Jacobian trust-region algorithm*, Computational Optimization and Applications, 48 (2011), pp. 255–271.
[48] A. Walther, S. R. R. Vetukuri, and L. T. Biegler, *A first-order convergence analysis of trust-region methods with inexact Jacobians and inequality constraints*, Optimization Methods and Software, 27 (2012), pp. 373–389.
[49] L. Wirsching, H. G. Bock, and M. Diehl, *Fast NMPC of a chain of masses connected by springs*, in Proceedings of the IEEE International Conference on Control Applications, Munich, 2006, pp. 591–596.